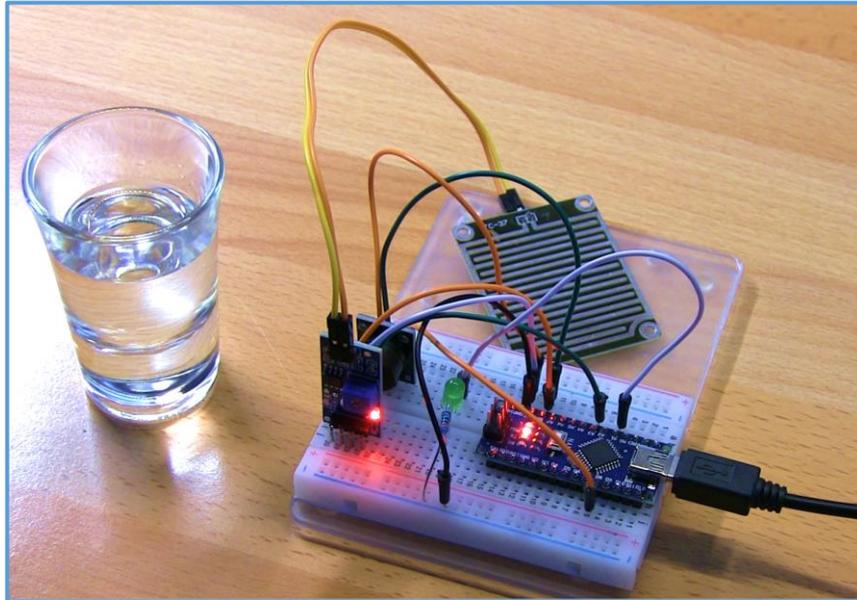


Heizkörper Wasser-Alarm mit Arduino NANO



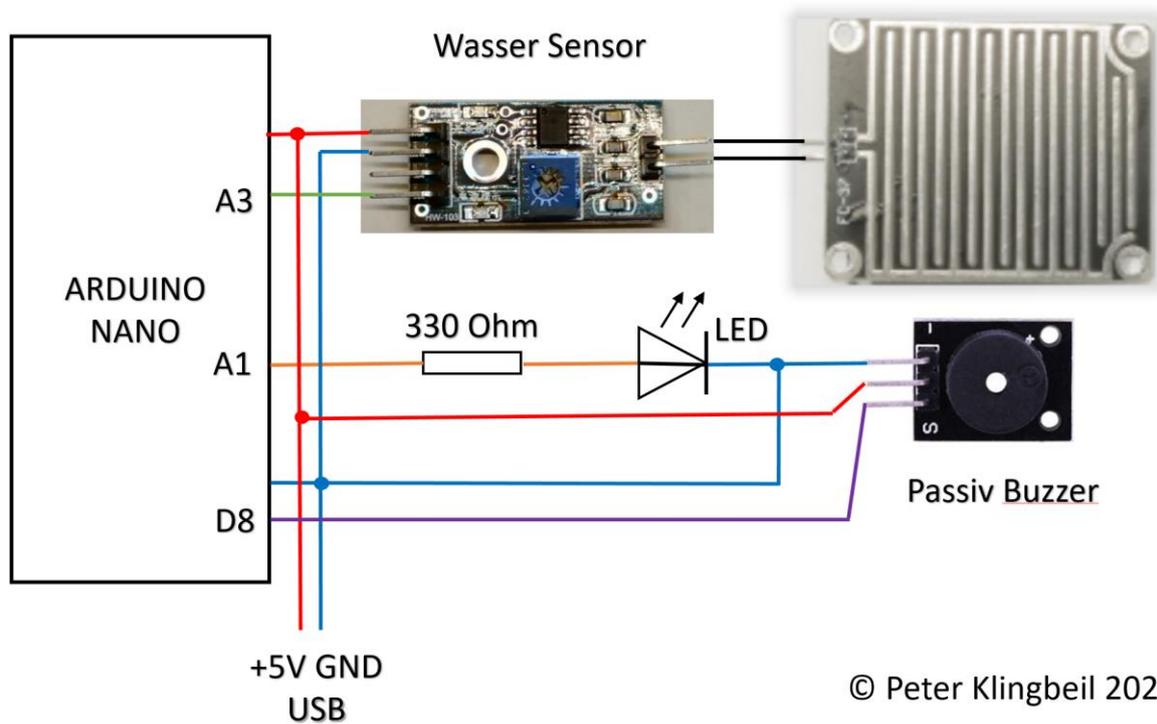
Vor kurzen leckte in meiner Wohnung ein alter Heizkörper und verlor Wasser. Ich habe es noch rechtzeitig bemerkt. Um Schäden zukünftig vorzubeugen, habe ich ein kleines Alarmgerät gebastelt.

Es wird unter den Heizkörper auf dem Boden platziert. Wenn der Heizkörper Wasser verliert, werden kleinste Spuren Wasser erkannt und ein akustischer Alarm ausgelöst.

Ursprünglich wollte ich das System mit einer Batterie betreiben, aber der Stromverbrauch ist mit 25 mA zu hoch um langfristig den Betrieb z.B. mit einer Blockbatterie zu gewährleisten. Um Energie zu sparen versetze ich den Arduino Nano aber eine Minute lang in den Schlafmodus. Vorher führt das Programm eine Feuchtmessung durch. Sollte auch nur ein Tropfen Wasser den Sensor berührt haben, wird ein akustischer Daueralarm ausgelöst. Der Alarm wird durch das Entfernen der Versorgungsspannung beendet. Vor einer Messung wird die LED für 100 ms angesteuert, um zu zeigen, dass das Programm ausgeführt wird. Während der „Schlafphase“ verbraucht das System ca. 10 mA. Gespeist wird es mit 5 Volt über die USB Schnittstelle vom Nano. Ein kleines USB-Netzteil versorgt das System. Der mittlere gemessene Energieverbrauch beträgt somit ca. 60 mW.

Folgende Bauteile werden benötigt:

- Arduino Nano V3.0 oder höher
- Regentropfen Sensor (Elektronik mit Sensorplatine)
- Passiv Buzzer
- LED
- 330 Ohm Widerstand
- Gehäuse oder Steckbrett



© Peter Klingbeil 2021

Das Programm ist sehr kurz gehalten:

```
// Radiator water damage monitoring (c) 2021 Peter Klingbeil.
// Every 60 seconds it is checked whether the sensor reports water.
// The alarm occurs after the delay at the latest.
// In the event of an alarm, a permanent acoustic signal sounds.
// It can only be switched off by disconnecting the supply voltage.

#include "pitches.h" // Tone library
#include "Narcoleptic.h" // A sleep library for Arduino (C) 2010 Peter Knight (Cathedrow)

#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

// notes for an alarm tone
int alarm[] = {NOTE_C5, NOTE_C6};

int duration = 300; // 300 milliseconds tone duration
int sensorValue;

void setup()
{
  // initialize serial communicaton - test
  Serial.begin(9600);
  pinMode(A1, OUTPUT); // alive LED
}

void loop()
{
  sensorValue = analogRead(A3); // Get sensor value
  Serial.println(sensorValue); // print out the value
}
```

```

// delay in between reads for stability & flash the LED if the System is alive

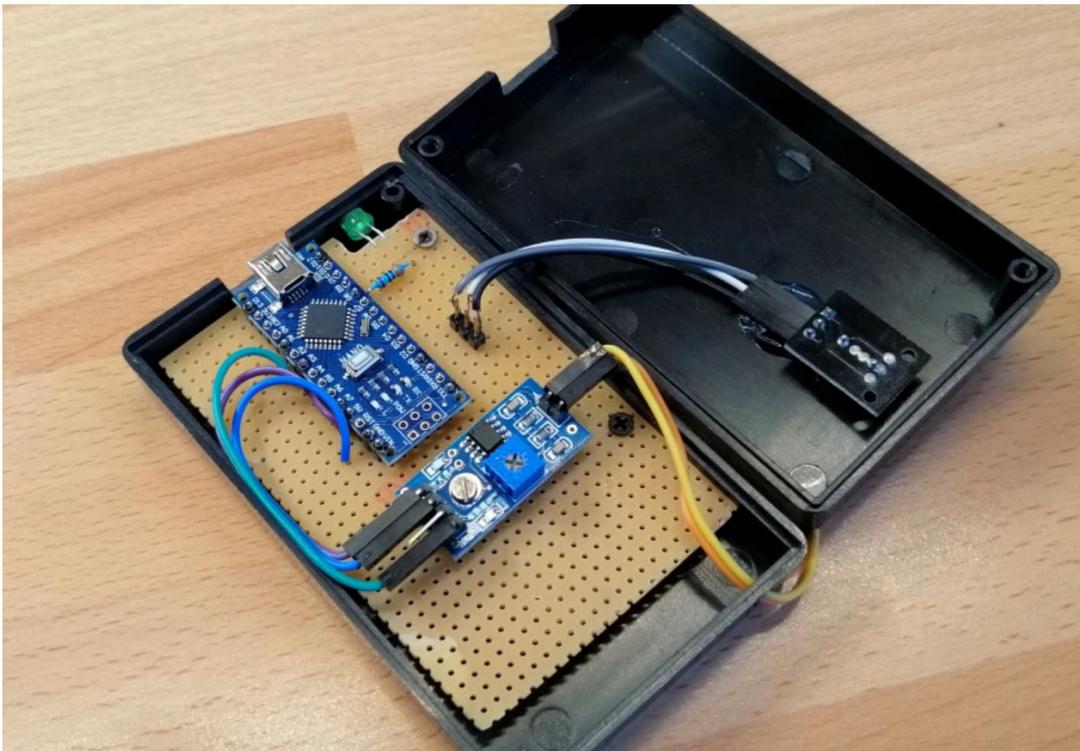
while (sensorValue < 1000) // Water!
{
  for (int thisNote = 0; thisNote < 2; thisNote++)
  {
    // pin8 output the voice, every scale is 0.1 sencond
    tone(8, alarm[thisNote], duration);
    // Output the voice after several minutes
    delay(100);
  }
}

digitalWrite(A1, HIGH); // turn the alive LED on
delay(100);           // wait (Puls)
digitalWrite(A1, LOW); // turn the alive LED off

// Set the CPU in sleep mode to reduce the power consumption from 25 mA to 9.5 mA
cbi(ADCSRA,ADEN); // disable A/C-Converter
pinMode(A1,INPUT); // set all used port to input to save power
Narcoleptic.delay(60000); // power down for 60 seconds
pinMode(A1,OUTPUT); // set all ports into state before sleep
sbi(ADCSRA,ADEN); // enable A/C-Converter
}

```

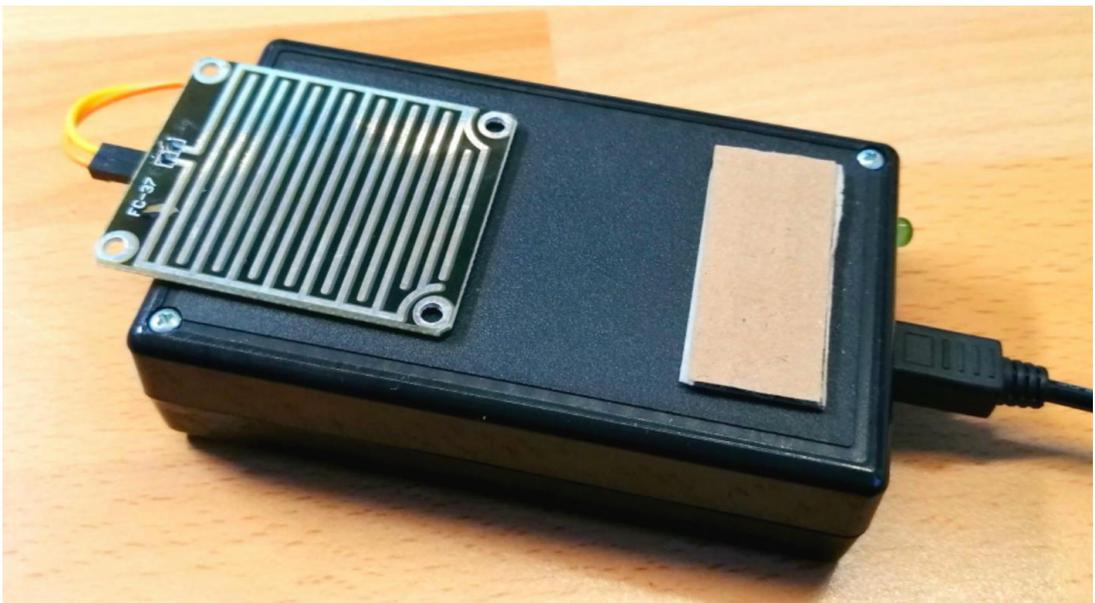
Das ganze System ist in ein kleines Gehäuse eingebaut.



Innenansicht



Oberseite



Rückseite

Den Programmcode und die benötigten Bibliotheken für die Arduino „IDE“ finden sich als ZIP-Datei auf meiner www.film-werk56.de Homepage unter LINUX.

Die Hardware kostete im Januar 2021 zusammen ca. 9 € + Gehäuse.

Viel Erfolg beim Nachbau, der Autor!